|  |
| --- |
| Reproducible Code Coursera Assignment #1 |

library(dplyr)

##   
## Attaching package: 'dplyr'

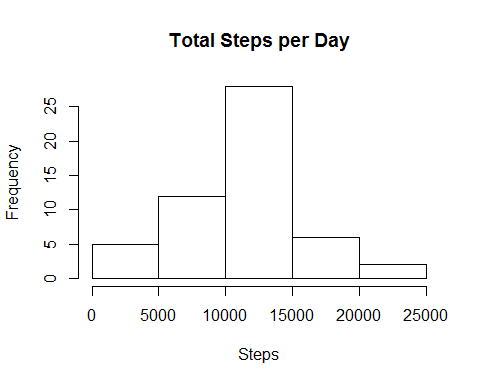
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
  
activity <- read.csv("activity.csv")  
  
activity$day <- weekdays(as.Date(activity$date))  
activity$DateTime<- as.POSIXct(activity$date, format="%Y-%m-%d")  
  
##Drop data without "NA"  
cleanData <- activity[!is.na(activity$steps),]

# Mean of total number of steps in a day

## Summary of total steps per day  
sumTable <- aggregate(activity$steps ~ activity$date, FUN=sum, )  
colnames(sumTable)<- c("Date", "Steps")  
  
## Plot Historgram for Steps per day  
hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day")



## Average of Steps  
as.integer(mean(sumTable$Steps))

## [1] 10766

## Median of Steps  
as.integer(median(sumTable$Steps))

## [1] 10765

The average number of steps per day = 10766 steps.

The median number of steps per day = 10765 steps.

# What is the average daily activity pattern?

Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

library(plyr)

## -------------------------------------------------------------------------

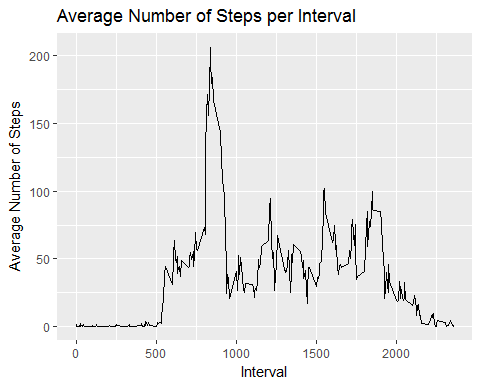
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(ggplot2)  
  
##Use data without 'NA'  
cleanData <- activity[!is.na(activity$steps),]  
  
##Create categories for steps  
intervalTable <- ddply(cleanData, .(interval), summarize, Avg = mean(steps))  
  
##Plot average steps per interval  
p <- ggplot(intervalTable, aes(x=interval, y=Avg), xlab = "Interval", ylab="Average Number of Steps")  
p + geom\_line()+xlab("Interval")+ylab("Average Number of Steps")+ggtitle("Average Number of Steps per Interval")



Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

##Maximum steps by category of intervals  
maxSteps <- max(intervalTable$Avg)  
  
##Which interval contains the maximum average number of steps  
intervalTable[intervalTable$Avg==maxSteps,1]

## [1] 835

The maximum number of steps for a 5-minute interval = 206 steps.

The 5-minute interval which had the maximum number of steps = 835 interval.

# Imputing missing values

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

##Find number of rows with NA in missing values  
  
nrow(activity[is.na(activity$steps),])

## [1] 2304

The total number of rows with steps = 'NA' is 2304.

Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

# Calculate a variable with average number of steps per weekday and interval  
avgTable <- ddply(cleanData, .(interval, day), summarize, Avg = mean(steps))  
  
## Create dataset with all 'NA's'   
nadata<- activity[is.na(activity$steps),]  
  
## Merging 'NA' data with average weekday interval for substitution  
newdata<-merge(nadata, avgTable, by=c("interval", "day"))  
  
## Reordering the newly substituded data in the same format as cleanData data set  
newdata2<- newdata[,c(6,4,1,2,5)]  
colnames(newdata2)<- c("steps", "date", "interval", "day", "DateTime")  
  
##Merge the 'NA' averages and non NA data together  
mergeData <- rbind(cleanData, newdata2)

Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

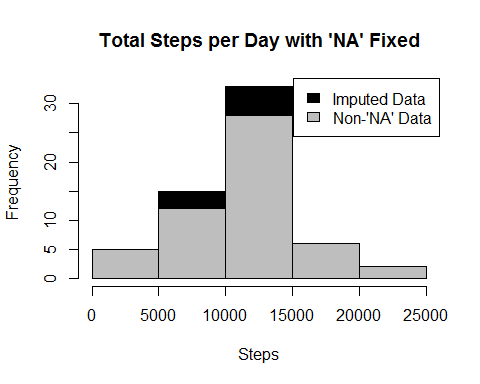
##Calculate sum of steps per date to compare with step 1  
sumTable2 <- aggregate(mergeData$steps ~ mergeData$date, FUN=sum, )  
colnames(sumTable2)<- c("Date", "Steps")  
  
## Average of Steps with 'NA' data considered in calculations  
as.integer(mean(sumTable2$Steps))

## [1] 10821

## Median of Steps with 'NA' data considered in calculations  
as.integer(median(sumTable2$Steps))

## [1] 11015

## Creating the histogram of total steps per day, categorized by data set to show impact  
hist(sumTable2$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with 'NA' Fixed", col="Black")  
hist(sumTable$Steps, breaks=5, xlab="Steps", main = "Total Steps per Day with 'NA' Fixed", col="Grey", add=TRUE)  
legend("topright", c("Imputed Data", "Non-'NA' Data"), fill=c("black", "grey") )



The new mean of the imputed data is 10821 steps compared to the old mean of 10766 steps. That creates a difference of 55 steps on average per day.

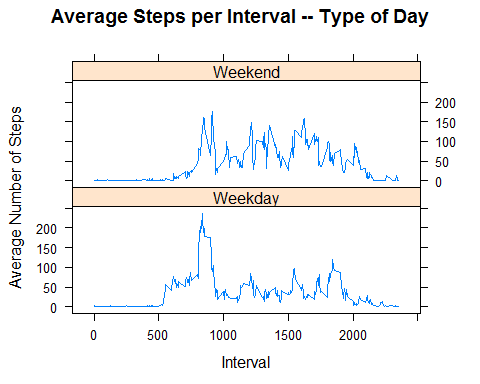
The new median of the imputed data is 11015 steps compared to the old median of 10765 steps. That creates a difference of 250 steps for the median.

However, the overall shape of the distribution has not changed.

# Are there differences in activity patterns between weekdays and weekends?

Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

# Create new category -- days of the week  
mergeData$DayCategory <- ifelse(mergeData$day %in% c("Saturday", "Sunday"), "Weekend", "Weekday")  
  
#Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).  
  
library(lattice)   
  
## Summarize data by interval and type of day  
intervalTable2 <- ddply(mergeData, .(interval, DayCategory), summarize, Avg = mean(steps))  
  
##Plot data in a panel plot  
xyplot(Avg~interval|DayCategory, data=intervalTable2, type="l", layout = c(1,2),  
 main="Average Steps per Interval -- Type of Day",   
 ylab="Average Number of Steps", xlab="Interval")

 Answer for the last question: Yes, the step activity trends vary based on whether the day falls on a weekend or not, possibly because of more time available during weekends.